A Flexible Material Database for Computational Science and Engineering

Josef Weinbub, Karl Rupp, Florian Rudolf
Institute for Microelectronics, Technische Universität Wien
weinbub@iue.tuwien.ac.at, rupp@iue.tuwien.ac.at,
rudolf@iue.tuwien.ac.at

Abstract

Simulation tools in the area of computational science and engineering usually require access to a plethora of material parameters for modeling physical phenomena. Academic software projects typically focus on a single approach for providing this capability. For instance, an extensible markup language (XML)-based input file is used, containing all relevant material parameters, such as the permittivity for silicon [1]. These parameters are then imported and accessed via XML libraries, like pugixml [2]. Although this approach is effective, it confines the application to use a single specific material database backend. For small simulation projects, this is hardly an issue. However, the larger the project and the higher the required usability, the more it becomes important to support different – typically file-based – material databases and to reuse historically-grown material database files. We tackle these challenges by using a flexible material database mechanism. Flexibility refers to decoupling a simulation tool from the actual material database kernel, such as XML. In particular, we strive for exchangeability and expandability via a variable database mechanism, based on a dynamic class hierarchy. We ensure proper handling of physical units via incorporating unit checks, which are especially important for robust numerical simulations [3]. Also, we investigate interfacing with Python, becoming increasingly important in the computational science and engineering community due to its support for rapid-prototyping. The developed methods are utilized in the free open source library ViennaMaterials [4]. We show the applicability of our approach based on a finite volume-based simulation tool due to its popular use in the area of semiconductor device simulation. The flexibility of switching database backends is shown, thus decoupling the simulation tool from a specific approach, such as XML. Based on the depicted examples, we show the advantage of applying our flexible material database over relying on a single specific backend.

References

1. M. Gayer and G. Iannaccone. A Software Platform for Nanoscale Device Simulation and Visualization. In Proc. of ACTEA, 2009.
2. A. Kapoulkine. pugixml. http://pugixml.org/.
3. B. Stroustrup. Software Development for Infrastructure. Computer 45(1), 2012.
4. J. Weinbub et al.. ViennaMaterials. https://github.com/viennamaterials/.